AF

JAN 24 2005 PATENT & TRADEMARK OFFICE

TRANSMITTAL OF APPEAL BRIEF (Large Entity)

Docket No. 872.0025.USU

In Re Application Of: **Jari A. Parviainen**

Application No.	Filing Date	Examiner	Customer No.	Group Art Unit	Confirmation No.
09/726,188	11/29/2000	Do, Chat C.	29683	2124	5987

Invention: **Dynamically Configurable Processor**

COMMISSIONER FOR PATENTS:

Transmitted herewith ~~in triplicate~~ is the Appeal Brief in this application, with respect to the Notice of Appeal filed on

The fee for filing this Appeal Brief is: **$500.00**

☒ A check in the amount of the fee is enclosed.

☐ The Director has already been authorized to charge fees in this application to a Deposit Account.

☒ The Director is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. **50-1924**

☐ Payment by credit card. Form PTO-2038 is attached.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

Signature

Dated: **January 19, 2005**

Robert J. Mauri (Reg. No. 41,180)
Harrington & Smith, LLP
4 Research Drive
Shelton, CT 06484-6212

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)] on 1/19/05 (Date)

Signature of Person Mailing Correspondence

Elaine F. Mian

Typed or Printed Name of Person Mailing Correspondence

cc:

IN THE U.S. PATENT AND TRADEMARK OFFICE

Appl. No.	:	09/726,188
Applicant	:	Jari A. Parviainen
Filed	:	November 29, 2000
TC/AU	:	2124
Examiner	:	Do, Chat C.
Docket No.	:	872.0025.USU
Customer No.	:	29683
Title	:	Dynamically Configurable Processor

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

Commensurate with the NOTICE OF APPEAL filed on November 24, 2004, Applicant/Appellant hereby submits this APPEAL BRIEF to the Board of Patent Appeals and Interferences (hereinafter, the Board) under 37 C.F.R. §41.31, and a draft for the $500 appeal brief fee set forth in 37 C.F.R. §41.20(b)(1). This BRIEF is filed within two months from the filing date of the above-cited NOTICE and the undersigned representative believes that no late fee is due. However, should the undersigned attorney be mistaken, please consider this a petition for an extension of time under 37 C.F.R. §1.136(a) or (b) that may be required to avoid dismissal of this appeal, and debit Deposit Account No. 50-1924 as appropriate.

(1) REAL PARTY IN INTEREST

The real party in interest (RPI) is Nokia Mobile Phones Limited, Finland, cited in an assignment of the U.S. application recorded on February 5, 2001.

(2) RELATED APPEALS AND INTERFERENCES

There are no other pending appeals or interferences of which the undersigned representative and assignee/RPI is aware that will directly affect, be directly affected by or have a bearing on the Board's decision in this appeal.

(3) STATUS OF CLAIMS

This application was filed on November 29, 2000 with 24 claims. Claims 1-24 are pending in this appeal, and are reproduced in an Appendix accompanying this Brief as those claims stood finally rejected by a final Office Action dated August 26, 2004.

(4) STATUS OF AMENDMENTS

No amendment to the claims was proposed subsequent to the final Rejection dated August 26, 2004.

(5) SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 is directed to a data processor (e.g., DSP core 10 of FIGS. 1A and 1B) that includes a multiplier block (e.g., 14 in FIGS. 1A and 6) having a multiplier front end (e.g., 14A in FIG. 6) for generating partial products from input operands. See page 6, lines 13-20; page 8, lines 16-24. The data processor further includes a plurality of Arithmetic Logic Units (ALUs) (e.g., the 8-bit or 16-bit ALU in ALU 14B in FIGS. 6 and 7) having inputs that are switchably coupled, in a first mode of

operation, to first data sources (e.g., buses X and Y in FIGS. 6 and 7) representing outputs of the multiplier front end. In the first mode of operation the ALUs add together partial products received from the multiplier front end to arrive at a multiplication result. See, e.g., FIG. 1A and corresponding text at page 6, lines 13-20. In a second mode of operation, the inputs of the plurality of ALUs are switchably coupled to second data sources (e.g., registers A and B 14F in FIG. 7) for performing at least one of arithmetic and logical operations on data received from the second data sources (e.g., registers A and B 14F in FIG. 7). See FIG. 1B and corresponding text at page 6, lines 21-27. See also, e.g., FIGS. 6 and 7 and corresponding text at page 8, line 16 to page 9, line 16, for exemplary descriptions of configurations of a multiplier 14 implementing the first and second modes of operation.

Independent claim 10 is directed to method for operating a data processor, such as the data processor in independent claim 1. The method is supported, for instance, by FIGS. 1A and 1B and corresponding text at page 6, lines 13-27; and FIGS. 6 and 7 and corresponding text at page 8, line 16 to page 9, line 16.

Independent claim 21 is directed to a Digital Signal Processor (DSP) comprising a DSP core (DSP core 10 of FIGS. 1A and 1B) having a register file e.g., register file 12 of FIGS. 1A and 1B), at least one ALU (e.g., ALUs 16 in FIGS. 1A and 1B), and at least one multiplier block (e.g., 14 of FIG. 6) comprised of a multiplier front end (e.g., 14A of FIG. 6) for generating partial products from input operands, said multiplier block (e.g., 14 of FIG. 6) further comprising circuitry for adding together said partial products (e.g., partial sums shown in FIG. 6), said circuitry comprising a plurality of ALUs (e.g., array of 8-bit ALUs 14B, each of which comprises an 8-bit ALU, in FIG. 6) having inputs that are programmably coupled, in a first mode of operation, to first data

sources (e.g., buses X and Y) comprised of outputs of said multiplier front end (e.g., 14A in FIG. 6) for adding together partial products (e.g., partial sums shown in FIG. 6) received therefrom to arrive at a multiplication result, said inputs of said plurality of ALUs being programmably coupled, in a second mode of operation, to second data sources (e.g., registers A and B 14F in FIG. 7) for selectively operating together in parallel for performing at least one of arithmetic and logical operations on data received from said second data sources (e.g., registers A and B 14F in FIG. 7), wherein said partial products have a width of n-bits, and where a width of individual ones of said plurality of ALUs is one of n-bits or less than n-bits. See, e.g., FIG. 7; page 4, lines 26-31; and page 9, lines 6-16.

(6) GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. The issue (Issue A) presented for review by the Board is whether claims 1, 2, 4, 7, 8, 10, 11, 13, 16, 17, 20, 21, and 24 are anticipated under 35 U.S.C. §102(e) by U.S. Patent No. 6,377,970 to Abdallah et al. (hereinafter, Abdallah). Under Issue A, claims 1, 7, 10, 16, and 20 stand or fall together, claims 2 and 11 stand or fall together, claims 21 and 24 stand or fall together, claims 4 and 13 stand or fall together, and claims 8 and 17 stand or fall together.

B. The second issue (Issue B) presented for review by the Board is whether claims 3, 5, 12, 14, 19, and 23 are obvious under 35 U.S.C. §103(a) by Abdallah in view of U.S. Patent No. 6,369,610 to Cheung et al. (hereinafter, Cheung). Under Issue B, claims 3, 5, 12, and 14 stand or fall together and claims 19 and 23 stand or fall together.

C. The third issue (Issue C) presented for review by the Board is whether claims 6, 9, 15, 18, and 22 are obvious under 35 U.S.C. §103(a) by Abdallah in view of

U.S. Patent No. 6,377,970 to Aldrich et al. (hereinafter, Aldrich). Under Issue C, claims 6 and 15 stand or fall together and claims 9, 18, and 22 stand or fall together.

(7) ARGUMENT

ISSUE A

Claims 1, 2, 4, 7, 8, 10, 11, 13, 16, 17, 20, 21, and 24 stand rejected as being anticipated under 35 U.S.C. §102(e) by Abdallah. As described above, under Issue A, claims 1, 7, 10, 16, and 20 stand or fall together, claims 2 and 11 stand or fall together, claims 21 and 24 stand or fall together, claims 4 and 13 stand or fall together, and claims 8 and 17 stand or fall together.

Claims 1, 7, 10, 16, and 20

Independent claims 1 and 10 recite a plurality of arithmetic logic units (ALUs). The plurality of ALUs have inputs switchably coupled to first and second data sources in first and second modes of operation, respectively. In the first mode of operation, the inputs are coupled to first data sources comprised of outputs of a multiplier front end for adding together partial products received therefrom to arrive at a multiplication result. In the second mode of operation, the inputs are coupled to second data sources for performing at least one of arithmetic and logical operations on data received from said second data sources.

The Examiner cites FIGS. 11 and 12 and associated text of Abdallah as anticipating each element of independent claims 1 and 10. For instance, the Examiner appears to equate the "carry-save adder (CSA) tree with carry lookahead adder (CLA)" 1110 and 1120 of FIG. 11 of Abdallah with the "ALUs" of independent claims 1 and 10. More specifically, the Examiner appears to equate outputs of the partial product generator

(e.g., 1100 of FIG. 11 of Abdallah) with the "first data sources" of independent claims 1 and 10 and the data input into 1140 of FIG. 11 of Abdallah with the "second data sources" of independent claims 1 and 10. See the paragraph spanning pages 7 and 8 of the final Office Action, dated August 26, 2004.

Applicant respectfully disagrees and traverses the §102(e) rejection of independent claims 1 and 10. First, Applicant states that an ALU is capable, for instance, of typical ALU functionality, such as addition and subtraction, as well as logical operations such as AND, OR, XOR and bit inversion, as well as shifting or load and store operations. See, e.g., page 7, line 30 to page 8, line 2 of Applicant's specification. The CSA tree with CLA 1110 and 1120 of Abdallah appear not to perform, e.g., any logical operation.

Nonetheless, even if the CSA tree with CLA 1110 and 1120 of Abdallah are assumed to meet an "ALU" as recited in independent claims 1 and 10, Applicant respectfully submits that not all elements of independent claims 1 and 10 are disclosed by Abdallah. For instance, FIG. 6 of Applicant's specification is illustrative of an ALU 14B having an 8-bit or 16-bit ALU. The 8-bit or 16-bit ALU has inputs coupled, in a first mode of operation, to first data sources (the X and Y buses) representing outputs of the multiplier front end. In a second mode of operation, the 8-bit or 16-bit ALU has inputs coupled to second data sources (registers A and B 14F, and the A-bus and B-bus, respectively). Further, Applicant states the following at page 8, lines 3-9 (emphasis added):

> When the multiplier 14 is operating in the multiplier mode the ALUs 16A are employed to add the partial sums as in the example of Fig. 4. The partial sums are output from a multiplier front-end, which may

> be constructed from AND gate logic as in Fig. 3. ***However, when operating as three 8-bit ALUs the inputs are provided instead from appropriate data buses, e.g., from two 8-bit buses***. The third ALU 16A may be used to, by example, perform some desired arithmetic or logical operation of the outputs of the two 8-bit ALUs, or on one of the results and another input.

Thus, Applicant discloses and claims ALUs having inputs that are switchably coupled for selecting one of the two separate data sources for the ALUs.

By contrast, Abdallah does not disclose ALUs having inputs that are switchably coupled for selecting one of the two separate data sources for the ALUs. For instance, the CSA tree with CLA 1110 and 1120, which the Examiner equates with "ALUs" of independent claims 1 and 10, are always connected to bus 1101 or 1102, respectively. There is no disclosure in Abdallah that any change is ever made to buses 1101 and 1101, and buses 1101 and 1102 to the CSA tree with CLA 1110 and 1120 (respectively) are therefore not switchably coupled to first and second data sources, as recited in independent claims 1 and 10. Even if buses 1101 and 1102 are "data sources" as recited in independent claims 1 and 10, bus 1101 is the sole data source for CSA tree with CLA 1110, and bus 1102 is the sole data source for CSA tree with CLA 1120.

The Examiner asserts that outputs of the partial product generator (the 16x16 multiplier using PADDH partial product selectors 1100 of Abdallah) equate with the "first data source" and the data input into the bus 1140 in FIG. 11 of Abdallah is the "second data source." The outputs of the partial product generator (the 16x16 multiplier using PADDH partial product selectors 1100 of Abdallah) are carried on buses 1101 and 1102 in Abdallah, so if the outputs of the partial product generator (the 16x16 multiplier

using PADDH partial product selectors 1100 of Abdallah) are considered to be the "first data source," then no "second data source" is disclosed in Abdallah, as there is no alternate to buses 1101, 1102.

Abdallah does disclose that when the signal CNTR2 is not asserted, a Packed Multiply-Add (PMAD) operation on packed data from buses 1140 and 1141 is performed. The block 1100 performs multiplication and creates 18-bit partial products, which are then added by the CSA tree with CLA 1110 and 1102. FIG. 12 of Abdallah provides a visualization of this multiplication, and partial products 1201 to 1028 are results from multiplying, e.g., A_0 of a packed data word A having $A_3A_2A_1A_0$ with B_0 of a packed data word B having $B_3B_2B_1B_0$. Each A_i and B_i appears to be 16 bits. See col. 9, line 24 to col. 10, line 5 (in particular, lines 28-29 of col. 9) and FIGS. 2 and 12 of Abdallah. When the signal CNTR2 is asserted, data from bus 1140 are inserted at certain positions in the partial products, as shown in FIG. 12 of Abdallah and described at col. 11, lines 18-30 of Abdallah. The data on bus 1140 is called G and has $G_7G_6G_5G_4G_3G_2G_1G_0$, where each G_i appears to be 8 bits. The data on bus 1141 is zero. What results is an addition of each of the packed data G_i in the packed word G. See FIG. 12 and col. 11, line 14 to col. 12, line 18 of Abdallah.

Although Abdallah discloses that bytes of the packed word G can be rearranged (see FIG. 12 of Abdallah), this is simply a rearranging of the data from bus 1140, on which G is received and not switchably coupling to another data source: Buses 1140, 1141, 1101, and 1102 are not changed. The Examiner appears to be equating data (e.g., outputs of the partial product generator and data input into the bus 1140 in FIG. 11 of Abdallah) with a "data source." However, in FIG. 7, Applicant shows an 8-bit (or 16-bit) ALU that is switchably coupled to two physically separate data sources. Moreover,

independent claims 1 and 10 recite that the inputs of said plurality of ALUs are switchably coupled, in a second mode of operation, to second data sources for performing at least one of arithmetic and logical operations on data ***received from*** said second data sources. Because operations are performed on data received from a data source, the data source is a physical item that can provide data.

For at least the reasons stated above, Abdallah does not disclose all features of independent claims 1 and 10 and Applicant respectfully submits that independent claims 1 and 10 are patentable over Abdallah. Because independent claims 1 and 10 are patentable, dependent claims 7, 16, 20 are also patentable for at least the reasons given with respect to independent claims 1 and 10.

Claims 2 and 11

Dependent claims 2 and 11 recite "wherein said partial products have a width of n-bits, and where a width of said ALUs is one of n-bits or less than n-bits." Applicant discloses, in an exemplary embodiment, a width of an ALU as being the width of the output of the ALU (e.g., where the width does not including a carry bit or carry bits). See, for instance, page 9, lines 11-14 and FIG. 6 of Applicant's specification.

If, as asserted by the Examiner, a CSA tree with CLA 1110, 1120 in Abdallah is considered to be an "ALU," then the "width" of one of the CSA tree with CLA 1110 and 1120 is 32 bits, while a partial product is 18 bits. For instance, Abdallah states the following at col. 9, lines 29-34 (emphasis added):

> The set of 16x16 multipliers 1100 multiply each packed data element Ai of the packed word data A received on the bus 1140 with the corresponding packed data element Bi of the packed word data B received

on the bus 1141 to produce ***thirty-two 18-bit partial products*** using radix 4 multiplication.

On the other hand, a CSA tree with CLA 1110, 1120 appears to be 32 bits wide, as it produces R00-R31. See col. 11, lines 17-31 (and in particular lines 30-31) and col. 9, lines 48-58 of Abdallah. If n=18 such that the width of the partial products in Abdallah is 18 bits, then the width (32 bits) of a CSA tree with CLA 1110, 1120 is greater than n=18 bits.

Furthermore, if one of the carry-save adders in a CSA tree with CLA 1110, 1120 is considered to be an "ALU" of Applicant's claims, the minimum number of output bits for a carry-save adder would be 20 bits. This can be seen in FIG. 12, where in order to add partial product 1201 (for example) with partial product 1202 (for example), a minimum of 20 bits is necessary for a result. Thus, a width of a carry-save adder in a CSA tree with CLA 1110, 1120 is greater than the width of a partial product.

Therefore, Abdallah does not disclose "wherein said partial products have a width of n-bits, and where a width of said ALUs is one of n-bits or less than n-bits," as recited in dependent claims 2 and 11, and dependent claims 2 and 11 are patentable over Abdallah.

Claims 21 and 24

Independent claim 21 recites the text of circuitry comprising a plurality of ALUs having inputs that are programmably coupled, in a first mode of operation, to first data sources comprised of outputs of a multiplier front end for adding together partial products received therefrom to arrive at a multiplication result, said inputs of said plurality of ALUs being programmably coupled, in a second mode of operation, to second data sources for selectively operating together in parallel for performing at least one of

arithmetic and logical operations on data received from said second data sources. As described above in reference to claims 1 and 10, there is no disclosure in Abdallah of inputs of a plurality of ALUs being programmably coupled, in first and second modes of operation, to first and second data sources. Furthermore, independent claim 21 comprises text of wherein said partial products have a width of n-bits, and where a width of individual ones of said plurality of ALUs is one of n-bits or less than n-bits. As described above in reference to claims 2 and 11, Abdallah does not disclose the text of wherein said partial products have a width of n-bits, and where a width of individual ones of said plurality of ALUs is one of n-bits or less than n-bits.

For at least these reasons, independent claim 21 is patentable over Abdallah. Because independent claim 21 is patentable over Abdallah, dependent claim 24 is patentable over Abdallah for at least the reasons given with respect to independent claim 21.

Claims 4 and 13

Dependent claims 4 and 13 recite "wherein said partial products have a width of 16-bits, and where a width of said ALUs is one of 16-bits, 8-bits or 4-bits." As described above in reference to claims 2 and 11, width of the partial products in Abdallah is 18 bits, and the width (32 bits) of a CSA tree with CLA 1110, 1120 is greater than n=18 bits. Therefore, Abdallah does not disclose "wherein said partial products have a width of 16-bits, and where a width of said ALUs is one of 16-bits, 8-bits or 4-bits" as recited in claims 4 and 13 and claims 4 and 13 are patentable over Abdallah.

Claims 8 and 17

Dependent claims 8 and 17 recite the text of "wherein said plurality of ALUs, when in the second mode of operation, operate in parallel with one another on data received from said second data sources." As described above in reference to independent claims 1 and 10, there is only one at most one "data source" for each CSA tree with CLA 1110, 1120 of Abdallah. Consequently, there is no disclosure of the text of "wherein said plurality of ALUs, when in the second mode of operation, operate in parallel with one another on data received from said second data sources," as recited in dependent claims 8 and 17. Claims 8 and 17 are therefore patentable over Abdallah.

ISSUE B

Claims 3, 5, 12, 14, 19, and 23 stand rejected as being obvious under 35 U.S.C. §103(a) by Abdallah in view of Cheung. Under Issue B, claims 3, 5, 12, and 14 stand or fall together and claims 19 and 23 stand or fall together.

Claims 3, 5, 12, and 14

Claims 3 and 12 recite the text of "wherein said partial products have a width of 8-bits, and where a width of said ALUs is one of 8-bits or 4-bits," while claims 5 and 14 recite the text of "wherein said partial products have a width of 32-bits, and where a width of said ALUs is one of 32-bits, 16-bits, 8-bits or 4-bits." As described above in reference to claims 2 and 11, width of the partial products in Abdallah is 18 bits, and the width (32 bits) of a CSA tree with CLA 1110, 1120 is greater than n=18 bits. Therefore, Abdallah does not disclose or imply the text in dependent claims 3, 5, 12, and 14. Applicant also cannot find disclosure or implication of the cited text in Cheung, as the partial products in Cheung appear to be determined though the use of an array of partial

calculating circuits, such that the partial products are either 4 or 5 bits wide. See Abstract and FIGS. 5 and 6 of Cheung. If neither Abdallah nor Cheung disclose or imply the text in dependent claims 3, 5, 12, and 14, then the combination of Abdallah and Cheung cannot disclose or imply this text.

Consequently, Applicant respectfully submits that dependent claims 3, 5, 12, and 14 are patentable over the combination of Abdallah and Cheung.

Claims 19 and 23

Dependent claims 19 and 23 further comprise the text of "wherein said plurality of ALUs comprise the same or additional ALUs coupled to inputs of said multiplier front end for changing a sign of said input operands." Applicant cannot find disclosure or implication of this text in Abdallah, and Abdallah appears to operate with only positive values for packed data. See col. 2, lines 42-54, col. 5, lines 51-59, and col. 7, lines 28-44 of Abdallah. Thus, Abdallah is not directed toward dealing with negative numbers, and there is no reason to combine Abdallah and Cheung, as Abdallah is unconcerned with negative numbers.

Meanwhile, Cheung states the following at col. 5, lines 7-19 (emphasis added):

> In the case of two's complement arithmetic, if one or both of the multiplicands is signed then the bits to be multiplied by the MSB of the ***other multiplicand can be inverted, and one added, using the additional logic present in the functional units 20*** (used in the bottom row and right hand column of the FAB), and 1 can be added to the output when necessary using the additional logic associated with the functional unit 32. This

> feature is under the control of control bits M_a (to cause the right hand column to operate as for signed arithmetic) and M_b (to cause the bottom row to operate as for signed arithmetic). In this way, products with one or both of the multiplicands being signed can be accommodated.

Thus, in Cheung, the array of partial calculating circuits (see FIG. 2 of Cheung) performs inversion, and the inversion appears not to occur by an ALU that is coupled to the inputs of a multiplier front end, as claimed in dependent claims 12 and 23. Because Abdallah is unconcerned with negative numbers and neither Abdallah nor Cheung disclose or imply the text of dependent claims 19 and 23 (such that the combination of Abdallah and Cheung cannot disclose or imply the text of dependent claims 19 and 23),Applicant submits that dependent claims 19 and 23 are patentable over the combination of Abdallah and Cheung.

ISSUE C

Claims 6, 9, 15, 18, and 22 stand rejected as being obvious under 35 U.S.C. §103(a) by Abdallah in view of Aldrich. Under Issue C, claims 6 and 15 stand or fall together and claims 9, 18, and 22 stand or fall together.

Claims 6 and 15

Dependent claims 6 and 15 contain text of "wherein said partial products have a width of n-bits, where a width of said ALUs is less than n-bits, and where at least some of said plurality of ALUs are switchably coupled together to provide an n-bit wide ALU." Applicant respectfully submits that Abdallah does not disclose or imply this text. Furthermore, Applicant submits that Aldrich also does not disclose or imply this text. While Aldrich does disclose two ALUs 378 and 380 (see FIG. 3 and col. 3, lines 41-50 of Aldrich), there is no disclosure or implication in Aldrich that ALUs 378 and 380 are

switchably coupled together to provide an n-bit wide ALU. Because neither Abdallah nor Aldrich disclose or imply the text in dependent claims 6 and 15, the combination of Abdallah and Aldrich cannot disclose or imply this text, and dependent claims 6 and 15 are patentable over the combination of Abdallah and Aldrich.

Claims 9, 18, and 22

Because independent claims 1, 10, and 21 are patentable, dependent claims 9, 18, and 22 are patentable for at least the reasons given above with respect to their respective independent claims.

The Applicant respectfully requests the Board reverse the final rejection in the Office Action of August 26, 2004, and further that the Board rule that the pending claims are patentable over the cited art.

Respectfully submitted:

HARRINGTON & SMITH, LLP

Robert J. Mauri
Reg. No.: 41,180

1/19/05
Date

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Claire F. Mann
Name of Person Making Deposit

1/19/05
Date

(8) CLAIMS APPENDIX

1. A data processor, comprising a multiplier block having a multiplier front end for generating partial products from input operands, and a plurality of arithmetic logic units (ALUs) having inputs switchably coupled, in a first mode of operation, to first data sources comprised of outputs of said multiplier front end for adding together partial products received therefrom to arrive at a multiplication result, said inputs of said plurality of ALUs being switchably coupled, in a second mode of operation, to second data sources for performing at least one of arithmetic and logical operations on data received from said second data sources.

2. A data processor as in claim 1, wherein said partial products have a width of n-bits, and where a width of said ALUs is one of n-bits or less than n-bits.

3. A data processor as in claim 1, wherein said partial products have a width of 8-bits, and where a width of said ALUs is one of 8-bits or 4-bits.

4. A data processor as in claim 1, wherein said partial products have a width of 16-bits, and where a width of said ALUs is one of 16-bits, 8-bits or 4-bits.

5. A data processor as in claim 1, wherein said partial products have a width of 32-bits, and where a width of said ALUs is one of 32-bits, 16-bits, 8-bits or 4-bits.

6. A data processor as in claim 1, wherein said partial products have a width of n-bits, where a width of said ALUs is less than n-bits, and where at least some of said plurality of ALUs are switchably coupled together to provide an n-bit wide ALU.

7. A data processor as in claim 1, wherein said inputs of said ALUs are switchably coupled under control of a program instruction.

8. A data processor as in claim 1, wherein said plurality of ALUs, when in the second mode of operation, operate in parallel with one another on data received from said second data sources.

9. A data processor as in claim 1, wherein said data processor forms a part of a wireless terminal.

10. A method of operating a data processor, comprising:

providing a multiplier block having a multiplier front end for generating partial products from input operands; and

providing said multiplier block with a plurality of arithmetic logic units (ALUs); wherein

in a first mode of operation, said plurality of ALUs have inputs switchably coupled to first data sources comprised of outputs of said multiplier front end for adding together partial products received therefrom to arrive at a multiplication result; and

in a second mode of operation, said inputs of said plurality of ALUs are switchably coupled to second data sources for performing at least one of arithmetic and logical operations on data received from said second data sources.

11. A method as in claim 10, wherein said partial products have a width of n-bits, and where a width of said ALUs is one of n-bits or less than n-bits.

12. A method as in claim 10, wherein said partial products have a width of 8-bits, and where a width of said ALUs is one of 8-bits or 4-bits.

13. A method as in claim 10, wherein said partial products have a width of 16-bits, and where a width of said ALUs is one of 16-bits, 8-bits or 4-bits.

14. A method as in claim 10, wherein said partial products have a width of 32-bits, and where a width of said ALUs is one of 32-bits, 16-bits, 8-bits or 4-bits.

15. A method as in claim 10, wherein said partial products have a width of n-bits, where a width of said ALUs is less than n-bits, and further comprising switchably coupling together at least some of said plurality of ALUs to provide an n-bit wide ALU.

16. A method as in claim 10, wherein said inputs of said ALUs are switchably coupled under control of a program instruction.

17. A method as in claim 10, wherein said plurality of ALUs, when in the second mode of operation, operate in parallel with one another on data received from said second data sources.

18. A method as in claim 10, wherein said data processor forms a part of a wireless terminal.

19. A method as in claim 10, wherein said plurality of ALUs comprise the same or additional ALUs coupled to inputs of said multiplier front end for changing a sign of said input operands.

20. A method as in claim 10, wherein switchably coupling employs reconfigurable signal routing logic.

21. A digital signal processor (DSP), comprising a DSP core having a register file, at least one arithmetic logical unit (ALU), and at least one multiplier block comprised of a multiplier front end for generating partial products from input operands, said multiplier block further comprising circuitry for adding together said partial products, said circuitry comprising a plurality of ALUs having inputs that are programmably coupled, in a first mode of operation, to first data sources comprised of outputs of said multiplier front end for adding together partial products received therefrom to arrive at a multiplication result, said inputs of said plurality of ALUs being programmably coupled, in a second mode of operation, to second data sources for selectively operating together in parallel for performing at least one of arithmetic and logical operations on data received from said second data sources, wherein said partial products have a width of n-bits, and where a width of individual ones of said plurality of ALUs is one of n-bits or less than n-bits.

22. A DSP as in claim 21, wherein said DSP forms a part of a wireless terminal.

23. A DSP as in claim 21, wherein said plurality of ALUs comprise the same or additional ALUs that are coupled to inputs of said multiplier front end for changing a sign of said input operands.

24. A DSP as in claim 21, and further comprising reconfigurable signal routing logic for providing data paths to and from said plurality of ALUs.

END OF CLAIMS